Docket No. SJO920030085US1
Firm No. 0037.0061

METHOD, SYSTEM, AND PROGRAM FOR RESTORING DATA TO A FILE

BACKGROUND

Field

[001] The present description relates to a method, system, and program for restoring data.

Description of Related Art

[002] There are various known techniques for backing up data. These backup techniques are often implemented using a storage-management server which can store data "objects" in one or more locations often referred to as storage pools. A data object stored by the server may be a "file" which is a collection of data or information that has a name, typically referred to as the "filename" and is stored in a file system which organizes the files. There are a number of different types of files including data files, program files, text files, directory files, etc. Different types of files are used to store different types of information. For example, program files store programs, whereas text files store text. The storage-management server frequently uses a database for tracking information about the stored objects, including the attributes and locations of the objects in the storage pools.

[003] The files to be backed up are usually located on one or more client stations coupled to the storage-management server over a network. One backup technique typically includes a "tape rotation" procedure, in which full, differential and incremental backups are made from a client station to a storage such as tape storage. Another approach which is used by some storage-management servers, such as the Tivoli Storage Manager ™ (TSM ™) product marketed by International Business Machines Corporation (IBM), utilizes a "progressive incremental" methodology, in which objects are backed up once from a client node and thereafter are typically not backed up again unless the object changes. In combination with the progressive incremental procedures, object-level policy

rules may be used to control the retention time and the number of versions which are maintained for stored objects. For example, the storage-management server can be configured to retain an "active" version, that is, an object currently residing on the client node, and a specified number of inactive versions, that is, objects that once resided on the client node but have since been deleted or modified.

[004] Should a user file originally stored on a client station become lost or corrupted, the backup copy of the file may be retrieved from the data storage subsystem and stored on the client station to replace the lost or corrupted file. In addition to user files, an "image" backup may also be performed which stores an image of the data stored on a "raw storage device." As used herein, a "raw storage device" refers to one or more storage units such as one or more "volumes."

[005] In computers, a volume is an identifiable unit of data storage that is sometimes physically removable from the computer or storage system. In tape storage systems, a volume may be a tape cartridge or a tape reel). In other storage systems, a volume may be a removable hard disk. Each volume typically has a system-unique name or number that allows it to be specified by a user. In some systems, a physical storage unit such as a magnetic disk drive may be divided into several separately identifiable volumes or separately identifiable partitions. Conversely, a volume may include more than one physical storage unit. For example, a Redundant Array of Independent Disks (RAID) volume may comprise several magnetic disk drives. Still further, the boundaries of a "logical volume" may not correspond to physical storage units. Instead, a logical volume may include storage units which are non-contiguous and may be scattered across several different physical storage units.

[006] To prepare a storage device for storing files, the device is typically "formatted" by the operating system. For example, if the storage device is a magnetic disk drive, the operating system may test the disk sectors to identify and mark any bad sectors. In addition, the operating system typically creates a file system which usually includes a directory to organize files and an address table to locate the files stored within

the disk drive. The file system directory and address table are typically stored within the storage device itself. Thus, if the formatted storage device is a volume, the file system directory and address table information identifying the location of the files are typically within the volume along with the user files stored within the volume.

[007] When performing an image backup of a raw storage device such as a volume, typically, the entire contents of the volume is read and stored by the storage-management server as image data representing the entire contents of the volume. Thus, the contents stored as image data include not just the files stored within the volume but also any file system including the file directories and address tables of the volume. Should the user files or the file system stored on the source volume become corrupted or lost, the image data stored by the storage-management server may then be used to restore the entire contents of the source volume including the user files and file system, back to a target volume which may be the same source volume or a different volume of the same or larger size.

[008] In addition to volumes, an image backup may be made of the contents of other types of raw storage devices including a disk slice or partition. A disk-shaped storage medium may be subdivided into physical portions, each of which is typically in the form of a concentric circle or track. Each track is typically divided into "sectors," each sector often representing the smallest unit of storage that is addressable for a read or write operation. Typically, a sector is 512 bytes in length. Thus, data can be located by identifying the concentric track and sector in which the data is stored.

[009] Further improvements in data storage and restoration may be useful in a variety of applications.

SUMMARY OF ILLUSTRATED EMBODIMENTS

[0010] Provided is a method, system and program for backing up the contents of a source storage device as an object in a data storage subsystem wherein the object contains image data representing the contents of the source storage device, and restoring the contents of the source storage device from the object to a file such as a flat file. The contents of the file may be copied to a target storage device to restore the contents of the source storage device from the file to the target storage device which may be the source storage device or another target storage device.

BRIEF DESCRIPTION OF THE DRAWINGS

[0011] Referring now to the drawings in which like reference numbers represent corresponding parts throughout:

FIG.1 illustrates an example of a computing environment in which aspects of the illustrated embodiments may be implemented;

FIG. 2 is a schematic diagram of a digital data processing machine of the embodiment of FIG. 1;

FIG. 3 shows an exemplary signal-bearing medium in accordance with described embodiments;

FIG. 4 is a schematic diagram showing the subcomponents of an illustrative storage hierarchy in accordance with described embodiments;

FIG. 5shows illustrative operations of an example of a data storage subsystem in accordance with described embodiments;

FIG. 6 illustrates an example of restoration of image backup data to a flat file in accordance with described embodiments; and

FIG. 7 illustrates an architecture of computing components in a network environment, such as the hosts, storage controllers, clusters, and any other computing devices.

DETAILED DESCRIPTION OF ILLUSTRATED EMBODIMENTS

[0012] In the following description, reference is made to the accompanying drawings which form a part hereof and which illustrate several embodiments. It is understood that other embodiments may be utilized and structural and operational changes may be made without departing from the scope of the present invention.

[0013] One aspect of the description provided herein concerns a storage management system, which may be embodied by various hardware components and interconnections. One example is shown by the storage management system 100 of FIG. 1. Broadly, the system 100 includes a data storage subsystem 102, one or more administrator stations 104, and one or more client stations 106. The subsystem 102 operates in response to directions of the client stations 106, as well as the administrator stations 104.

[0014] The administrator stations 104 are used by system administrators to configure, monitor, and repair the subsystem 102. Under direction of an end user, the client stations 106 use the subsystem 102 to store and manage data on their behalf. More particularly, each client station 106 creates and regards data in the form of "user files". In this regard, each client station 106 separately employs the subsystem 102 to archive, backup, retrieve, and restore its user files. Accordingly, each user file is associated with a single client station 106, which is the source of that user file. In addition to backups of user files, image backups may be made for raw storage devices of each client station 106. As previously mentioned, a raw storage device may be a storage unit such as volume, partition, or disk slice, for example.

[0015] Each client station 106 may comprise any general purpose computer, such as an RS-6000 based workstation, Intel processor-based personal computer, mainframe computer, etc. The client stations 106 may comprise similar or different machines, running the similar or different operating systems. Some exemplary operating systems include AIX™, UNIX, OS/2, WINDOWS™, etc.

[0016] The client stations 106 are interconnected to the subsystem 102 by a network 116. The network 116 may comprise any desired connection, including one or more

conductive wires or busses, fiber optic lines, data communication channels, wireless links, internet connections, telephone lines, Storage Area Network (SAN), Local Area Network (LAN), Intranet, the Internet, Wide Area Network (WAN), etc. Preferably, a high speed communication channel such as a T3 link is used, employing a network protocol such as TCP/IP.

[0017] The administrator stations 104 comprise electronic equipment for a human or automated storage administrator to convey machine-readable instructions to the subsystem 102. Thus, the stations 104 may comprise processor-equipped general purpose computers or "dumb" terminals, depending upon the specific application.

[0018] In an exemplary embodiment, the data storage subsystem 102 may comprise a commercially available server such as an the Tivoli Storage Manager ™ (TSM ™) product distributed by IBM, which has been modified to operate in accordance with the description provided herein. However, since other hardware arrangements may be used as well, a generalized view of the subsystem 102 is discussed below.

[0019] The data storage subsystem 102 includes a data processing apparatus 108, having a construction as discussed in greater detail below. The data processing apparatus 108 exchanges signals with the network 116 and the client stations 106 via an interface 112, and likewise exchanges signals with the administrator stations 104 via an interface 110. The interfaces 110, 112 may comprise any suitable device for communicating with the implemented embodiment of client station and administrator station. For example, the interfaces 110, 112 may comprise ETHERNET cards, small computer system interfaces ("SCSIs"), parallel data ports, serial data ports, telephone modems, fiber optic links, wireless links, etc.

[0020] The data processing apparatus 108 is also coupled to a database 113 and a storage hierarchy 114 in which image data representing the contents of a raw storage device may be stored in one or more objects as records of in the database 113. In one aspect of the description provided herein, the data storage subsystem 102 can restore the image data, using the database objects, to a file in a file system, as well as to a target raw

storage device, depending upon the particular application. As explained in greater detail below, such an arrangement may facilitate data restoration in a variety of circumstances.

[0021] In addition to image data, the storage hierarchy 114 is used to store selected user files from the client stations 106. The user files may be stored as individually or aggregated into managed files. The subsystem's storage of image data and user files may protect the contents of storage units on a client's machine from loss or corruption, assist the clients by freeing storage space at the client stations, and may also provide management of client data. In this respect, operations of the storage hierarchy 114 may include "archiving" data from the client stations 106, "backing up" data of the client stations 106 contained in the storage hierarchy 114, "retrieving" stored data for the client stations 106, and "restoring" data backed-up on the hierarchy 114.

[0022] The database 113 contains information about the image data objects and user files contained in the storage hierarchy 114. . For example, as described in U.S. Pat. No. 6,098,074, this information may include the addresses at which image data objects are stored in the database in the storage hierarchy 114, various characteristics of the stored data, certain client-specified data management preferences, etc. The client information includes information relative to the client station 106 with which the image data is associated. In the one example, the client information is represented by "client number", "client type", and "source". For an image data object, the "client number" identifies the originating client station 106. This identification may include a numeric, alphabetic, alphanumeric, or other code. In this example, a numeric code is shown. The "client type" associates the client with one or more predetermined categories, such as different computer types, operating systems, communications parameters, etc. The "source" column lists a location in the client station 106 where the image data is stored locally by the client. As a specific example, an image source may comprise a partition of a storage drive in the client station.

[0023] The data processing apparatus 108 may be embodied by various hardware components and interconnections. FIG. 2 shows one example, in the form of a digital data processing apparatus 200.

[0024] The apparatus 200 includes a processing unit 202, such as a microprocessor or other processing machine, coupled to a storage unit 204. In the present example, the storage unit 204 includes a fast-access storage 206 as well as nonvolatile storage 208. The fast-access storage 206 preferably comprises random access memory, and may be used to store programming instructions executed by the processing unit 202. The nonvolatile storage 208 may comprise, for example, one or more magnetic data storage disks such as a "hard drive", a tape drive, or any other suitable physical storage device. The apparatus 200 also includes at least one input/output 210, such as a line, bus, cable, electromagnetic link, or other means for exchanging data between the processing unit 202 and other components of the subsystem 102.

[0025] Despite the specific foregoing description, ordinarily skilled artisans (having the benefit of this disclosure) will recognize that the apparatus discussed above may be implemented in a machine of different construction, without departing from the scope of the description provided herein. As a specific example, one of the components 206 or 208 may be eliminated; furthermore, the storage unit 204 may be provided on-board the processing unit 202, or even provided externally to the apparatus 200.

[0026] The storage hierarchy 114 may be implemented in storage media of various number and characteristics, depending upon the clients' particular requirements. To specifically illustrate one example, FIG. 4 depicts a representative storage hierarchy 400. The hierarchy 400 includes multiple levels 402-410, where successively higher levels represent incrementally higher storage performance. The levels 402-410 provide storage devices with a variety of features and performance characteristics.

[0027] In this example, the first level 402 includes high-speed storage devices, such as magnetic hard disk drives, writable optical disks, or other direct access storage devices ("DASDs"). The level 402 provides the fastest data storage and retrieval time among the

levels 402-410, albeit the most expensive. The second level 404 includes DASDs with less desirable performance characteristics than the level 402, but with lower expense. The third level 406 includes multiple optical disks and one or more optical disk drives. The fourth and fifth levels 408-410 include even less expensive storage means, such as magnetic tape or another sequential access storage device.

[0028] The levels 408-410 may be especially suitable for inexpensive, long-term data archival, whereas the levels 402-406 are appropriate for short-term fast access data storage. As an example, one or more devices in the level 402 and/or level 404 may even be implemented to provide a data storage cache.

[0029] Devices of the levels 402-410 may be co-located with the subsystem 102, or remotely located, depending upon the user's requirements. Thus, storage devices of the hierarchy 400 may be coupled to the data processing apparatus 108 by a variety of means, such as one or more conductive wires or busses, fiber optic lines, data communication channels, wireless links, internet connections, telephone lines, SCSI connection, ESCON connect, etc.

[0030] Although not shown, the hierarchy 400 may be implemented with a single device type, and a corresponding single level. Ordinarily skilled artisans will recognize the "hierarchy" being used illustratively, since the description provided herein includes but does not require a hierarchy of storage device performance.

[0031] In the context of the storage hierarchy 114/400, the term "storage pool" is used to identify one or more storage devices with similar performance characteristics. For instance, the level 404 may be comprised of several storage pools, each pool including one or more DASDs.

[0032] FIG. 5 shows an example of operations of a data storage subsystem 102 which can restore image data from a source raw storage device to a file as well as to a target raw storage device, to facilitate data restoration operations. The image data is initially created by backing up (block 520) a source raw storage device to separate media. In the example of FIG. 5, the source raw storage device is a logical volume of one or more

magnetic disk drives of a client station 106. The source logical volume is schematically represented by a volume 622 in FIG. 6. However, it should be appreciated that the source raw storage device may be other than a logical volume such as a partition, or disk slice as described above. The image data backed up from the source logical volume is stored in a database as one or more objects tracked by the database on a suitable storage unit 624 of the storage hierarchy 114. This may be a magnetic disk drive or tape drive, for example.

[0033] In one embodiment, prior to commencing the image backup operation, the source logical volume may be unmounted and then remounted as read-only so that the source logical volume is mounted but unavailable for write operations. Write operations to the source logical volume during the image backup may cause corruption of the contents of the source logical volume in some applications. Following completion of the image backup, the source volume may be remounted to permit write operations to the source logical volume. Alternatively, in some applications, by refraining from writing to the source logical volume during an image backup, unmounting and remounting may be reduced or eliminated.

[0034] In yet another embodiment, the image may be backed up to an object in the database in which the first object is designated the "original" object. Any changes to the data of the source volume during the image backup can be backed up to a second object in the database in which the second object is designated the "updated" object. Such an arrangement facilitates image backups without unmounting the source volume.

[0035] In one aspect of the description provided herein, a restore operation utilizing the backup image data may be commenced without a target logical volume. For example, in a number of applications it may be useful to the backup image data contained within the database object or objects to a file prior to any attempt to restore the backup image data using the objects to a target logical volume. Thus, in the operations of FIG. 5, an inquiry is made (block 528) as to whether the restore operation is to be directed to a target logical volume. If not, the database objects image data representing the contents of the source logical volume 622 may be used to restore (block 530) the source volume

contents directly to a flat file 632 as shown in FIG. 6. The restoration process uses the database objects containing the image data. As a result, the flat file 632 can contain the entire contents of the source logical volume 622 including the formatted file system of the source logical volume 622. If the source logical volume 622 contained only raw unformatted data without a file system, the flat file 632 can contain all of the raw data of the source logical volume 622.

[0036] In some applications, the flat file 632 may provide a temporary repository for the contents of the source volume 622. In other applications, a file such as the flat file 632 may be actively used. For example, in the LINUX operating system, a file may be mounted as a storage unit. Thus, the files stored within the flat file may be accessed for use by an administrator station 104 or a client station 106, for example, using the file system contained within the flat file 622.

[0037] In some applications, following an image backup of a source raw storage device, an image restore operation may be directed to a target raw storage device of a client station 106. This may occur for example, when the source raw storage device fails or the contents on the source raw device becomes corrupted or otherwise unavailable. Also, in some applications it may be appropriate to create a mirror image of the source raw device on a target raw storage device.

[0038] In the example of FIGs. 5 and 6, the source raw storage device is the source logical volume 622. If an image restore operation is to be directed (block 528) to a target logical volume, a determination (block 540) is made as to whether a suitable target logical volume is available. The target logical volume may be the source logical volume 622 itself or may be another target logical volume. Typically, the target raw storage device will have a size at least as large as the source raw storage. In the example of FIGs. 5 and 6, the target raw storage device is represented by a target logical volume 642 of a client station 106. If the target logical volume 642 is available (block 540) the entire contents of the source logical volume 622 may be restored to the target logical volume 642 in a conventional image restore operation (block 544) using the database objects

containing the backup image data stored on the storage unit 624 in the storage hierarchy 114. For example, the backup image data stored on the storage unit 624 may be restored to the target logical volume 642 using the database objects in the manner utilized by the commercially available Tivoli Storage Manager™.

[0039] However, a target logical volume 642 may not be available (block 540) for a variety of reasons. For example, the client station 106 hosting the target logical volume 622 may be unreachable from the data storage subsystem 102 or the traffic on the network 116 may be too slow for a large restore operation. If so, the image data representing the contents of the source logical volume 622 may be restored (block 530) directly to a flat file 632 as discussed above, using the database objects containing the backup image data.

[0040] Still further, following an image restore operation to a target logical volume 642, it may be determined that some or all of the contents has been corrupted. This corruption may caused by a variety of factors. For example, the source logical volume 622 may have been already corrupted at the time of the image backup operation (block 520). Alternatively, the image restore operation (block 544) to the target logical volume 642 may have had an error. Yet another possibility is that the target logical volume had physical problems such as bad sectors which corrupted the data.

[0041] Thus, where the contents of the source logical volume 622 as restored to the target logical volume 642 contains corrupted data, the target logical volume 642 can be thought of as unavailable (block 540), and the image data stored on the storage device 624 in the form of database objects and representing the contents of the source logical volume 622, may be restored (block 530) directly to a flat file 632 as discussed above.

[0042] As previously mentioned, following the restore operation (block 530) to the flat file 632, the flat file 632 will contain the entire contents of the source logical volume 622 including the formatted file system (if any) of the source logical volume 622. Alternatively, if the source logical volume 622 contained only raw unformatted data

lacking a file system, the flat file 632 will contain all of that raw data of the source logical volume 622.

[0043] Once a target logical volume becomes available (block 550), the contents of the flat file 632 may be copied (block 552) in a simple copy operation to a target logical volume such as the volume 654 (FIG. 6) which may be the same as the source logical volume 622 or a different logical volume. For example, the contents of the flat file 632 may be copied to the target logical volume 654 using the Unix "dd" copy command. Copy commands in other operating systems may be used as well.

[0044] Thus, if the target logical volume is unavailable (block 540) because, for example, the client station 106 hosting the target logical volume 622 or 642 is unreachable from the data storage subsystem 102 or the traffic on the network 116 is too slow for a large restore operation, the backup image data on the storage device 624 may be restored (block 530) using the database objects, to a flat file 632 at a different client station 106. The flat file 632 may be compressed if appropriate and stored on a portable medium such a compact disk or a tape cartridge and carried to the client station 106 hosting the target logical volume 622 or 642. The flat file 632 may then be uncompressed and copied to the target volume 622 or 642 by the client station hosting the target logical volume. In this manner, the entire contents of the source logical volume 520, including any file system contained within the source logical volume 520, may be restored to a target logical volume, via a file 632.

[0045] As another example, if the target logical volume 642 contains corrupted data following a restore operation (block 544) in which the backup image data on the storage device 624 is restored to the target logical volume 642 using the database objects, a flat file 632 may be used to assist in diagnosing the problem. For example, the backup image data may also be restored (block 530) to a flat file 632 using the database objects, and the flat file may be copied (block 552) to another target logical volume 654. If it is determined that the target logical volume 654 likewise contains corrupted data, it may be determined that the source logical volume 622 likely contained corrupted data prior to the

image backup operation (block 520). If the target logical volume 654 is free of corrupted data, it may be determined that the source logical volume 622 was free of corrupted data and that either the target logical volume 642 likely contains one or more bad sectors or that an error occurred when the backup image data contained within the database objects stored on the storage device 624 was restored (block 544) to the target logical volume 642. Accordingly, the image restoration operation (block 544) may be repeated with another target logical volume. If the other target logical volume is determined to be free of data corruption after the backup image data is restored to the other target volume, it may be determined that the target logical volume 642 likely contains one or more bad sectors.

[0046] A "flat file" is a file containing one or more records that have no structured relationship amongst themselves. In the illustrated embodiment of FIG. 6, the flat file 632 contains a single record, that is, the image data of the entire contents of the source logical volume 622. It is appreciated that a flat file may contain multiple records wherein each record contains the image data of the contents of a different logical volume or other source raw storage device. It is further appreciated that the image data may be stored in a record of a file having other records and that such other records may have a structured relationship.

[0047] The flat file 632, as a file, can be assigned a filename, and can be stored in a storage device of a client station 106 or a storage device of the data subsystem 102. The storage device such as a volume which stores the flat file 632, can have a file system including a directory which identifies the name of the flat file 632 and organizes the files volume. The volume storing the flat file 632 may also have an address table which identifies the physical location of the flat file 632 within the volume storing the flat file 632.

Additional Implementation Details

[0048] The described techniques for managing resources may be implemented as a method, apparatus or article of manufacture using standard programming and/or

engineering techniques to produce software, firmware, hardware, or any combination thereof. The term "article of manufacture" as used herein refers to code or logic implemented in hardware logic (e.g., an integrated circuit chip, Programmable Gate Array (PGA), Application Specific Integrated Circuit (ASIC), etc.) or a computer readable medium, such as magnetic storage medium (e.g., hard disk drives, floppy disks,, tape, etc.), optical storage (CD-ROMs, optical disks, etc.), volatile and non-volatile memory devices (e.g., EEPROMs, ROMs, PROMs, RAMs, DRAMs, SRAMs, firmware, programmable logic, etc.). Code in the computer readable medium is accessed and executed by a processor complex. The code in which preferred embodiments are implemented may further be accessible through a transmission media or from a file server over a network. In such cases, the article of manufacture in which the code is implemented may comprise a transmission media, such as a network transmission line, wireless transmission media, signals propagating through space, radio waves, infrared signals, etc. Thus, the "article of manufacture" may comprise the medium in which the code is embodied. Additionally, the "article of manufacture" may comprise a combination of hardware and software components in which the code is embodied, processed, and executed. Of course, those skilled in the art will recognize that many modifications may be made to this configuration without departing from the scope of the present invention, and that the article of manufacture may comprise any information bearing medium known in the art.

[0049] For example, in the context of FIGS. 1-2 the method aspect of the description provided herein may be implemented, by operating the data processing apparatus 108 (embodied by a digital data processing apparatus 200), to execute a sequence of machine-readable instructions. These instructions may reside in various types of signal-bearing media. In this respect, one aspect of the present description concerns a programmed product, comprising signal-bearing media tangibly embodying a program of machine-readable instructions executable by a digital data processor to perform a method of

storing image data and restoring image data to a file and copying the file to a target device.

[0050] Illustratively, this signal-bearing media may comprise RAM contained within the data processing apparatus 108, as represented by the fast-access storage 206 for example. Alternatively, the instructions may be contained in another signal-bearing media, such as a magnetic data storage diskette 300 (FIG. 3), directly or indirectly accessible by the processing unit 202. Whether contained in the digital data processing apparatus 200 or elsewhere, the instructions may be stored on a variety of machine-readable data storage media, such as DASD storage (e.g., a conventional "hard drive" or a RAID array), magnetic tape, electronic read-only memory (e.g., ROM, EPROM, or EEPROM), an optical storage device (e.g. CD-ROM, WORM, DVD, digital optical tape), paper "punch" cards, or other suitable signal-bearing media including transmission media such as digital and analog and communication links and wireless. In an illustrative embodiment of the invention, the machine-readable instructions may comprise software object code, compiled from a language such as C, C++, PLX, etc.

[0051] The illustrated logic of FIGs. 5, 6 show certain events occurring in a certain order. In alternative implementations, certain operations may be performed in a different order, modified or removed. Moreover, operations may be added to the above described logic and still conform to the described implementations. Further, operations described herein may occur sequentially or certain operations may be processed in parallel. Yet further, operations may be performed by a single processing unit or by distributed processing units.

[0052] FIG. 7 illustrates one implementation of a computer architecture 700 of the network components, such as the data storage subsystem 102, administrator stations 104 or client stations 106 shown in FIG. 1. The architecture 700 may include a processor 702 (e.g., a microprocessor), a memory 704 (e.g., a volatile memory device), and storage 706 (e.g., a non-volatile storage, such as magnetic disk drives, optical disk drives, a tape drive, etc.). The storage 706 may comprise an internal storage device or an attached or

network accessible storage. Programs in the storage 706 are loaded into the memory 704 and executed by the processor 702 in a manner known in the art. The architecture further includes a network card 708 to enable communication with a network. An input device 710 is used to provide user input to the processor 702, and may include a keyboard, mouse, pen-stylus, microphone, touch sensitive display screen, or any other activation or input mechanism known in the art. An output device 712 is capable of rendering information transmitted from the processor 702, or other component, such as a display monitor, printer, storage, etc.

[0053] The foregoing description of various implementations of the present disclosure has been presented for the purposes of illustration and description. It is not intended to be exhaustive or to limit the present description to the precise form disclosed. Many modifications and variations are possible in light of the above teaching. It is intended that the scope be limited not by this detailed description, but rather by the claims appended hereto.